Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

Claim 1 (currently amended): An execution unit for use in a computer system for conditionally carrying out an operation defined in a computer instruction, the execution unit comprising:

first and second input stores for holding respective first and second source operands on which ~~an~~ the operation defined in the instruction is to be carried out, wherein each input store ~~defines~~ holds a plurality of objects of a predetermined size, each object defining one of a plurality of lanes, a maximum number of lanes being determined by a smallest allowable predetermined object size;

a plurality of operators associated respectively with said lanes for carrying out ~~an~~ the operation specified in the instruction on objects in corresponding lanes of said first and second input stores;

a destination buffer for holding the results of the operation on a lane-by-lane basis; and

selecting means for determining independently for each lane, in dependence on stored condition values derived from the results of executing a prior instruction sequence, whether or not the operation is to be executed on objects in that lane;

wherein a number of stored condition values corresponds to said maximum number of lanes in each of said first and second input stores, a prior operation being operable to generate said condition values so that, when the source operands have less than the maximum number

of lanes, two or more condition values are set to a same value so that each individual condition value is generated regardless of a degree of packing of the first and second source operands.

Claim 2 (original): An execution unit according to claim 1, wherein said condition values comprise a set of condition codes.

Claim 3 (original): An execution unit according to claim 2, wherein the selecting means comprises means for comparing selected ones of said set of condition codes with a test code identified in the instruction.

Claim 4 (previously presented): An execution unit according to claim 2, wherein the number of condition codes in said set corresponds to the maximum number of lanes in each of the first and second source operands.

Claim 5 (currently amended): An execution unit according to claim 2, which comprises a condition code generator for generating said set of condition codes responsive to execution of ~~an~~ the instruction.

Claims 6-7 (cancelled)

Claim 8 (currently amended): A computer system for conditionally carrying out an ~~operations~~ operation defined in a computer instruction, the computer system comprising:

fetch and decode circuitry for fetching and decoding a sequence of instructions from a program memory;

at least one execution unit according to any of claims 1 to 5; and

at least one memory access unit for effecting memory access operations responsive to memory access instructions.

Claim 9 (previously presented): A computer system according to claim 8, which comprises a condition code register for holding said condition values in the form of a set of condition codes.

Claim 10 (currently amended): A computer system according to claim 8, which includes a test register for holding a test code, the test register being addressed by [[a]] the computer instruction and said test code being used in comparison with said condition values to determine for each lane whether or not the operation is to be executed on objects in that lane.

Claim 11 (currently amended): A method of executing instructions on operands containing a plurality of packed objects, the method comprising:

accessing at least one source operand containing a plurality of packed objects in a plurality of lanes, each packed object having a predetermined size and defining one of said plurality of lanes, a maximum number of said lanes being determined by a smallest allowable predetermined object size;

accessing stored condition values derived from the results of executing a prior instruction sequence, to determine independently for each respective lane whether or not an operation defined in ~~the~~ an instruction is to be implemented on that lane of the source operand; and

carrying out the operation and updating a destination operand only in those lanes for which the stored condition value indicates that the operation should be implemented;

wherein a number of stored condition values corresponds to said maximum number of lanes in ~~said at least one input store~~ the source operand, a prior operation being operable to set condition values so that, when the source operands have less than a maximum number of lanes, two or more condition values are set to a same value so that each individual condition value is generated regardless of a degree of packing of the ~~first and second~~ source operands.

Claim 12 (previously presented): A method according to claim 11, wherein the stored condition values comprise a set of condition codes, held in a condition code register, and wherein the step of accessing the stored condition values comprises accessing said set of condition codes and comparing said condition codes with a test code identified in the instruction.

Claim 13 (previously presented): A method according to claim 12, wherein the test code is held in a test register which is identified by an address in the instruction.

Claim 14 (previously presented): An execution unit according to claim 5, wherein said condition codes comprise at least one of a negative flag, zero flag, carry flag, and overflow flag.

Claim 15 (currently amended): A method according to claim ~~12~~ 12, wherein said condition codes comprise at least one of a negative flag, zero flag, carry flag, and overflow flag.